Chapter 20
Theories, Models, Programs, and Tools of Design: Views from Artificial Intelligence, Cognitive Science, and Human-Centered Computing

Ashok K. Goel and Michael E. Helms

20.1 Introduction

Research on design adopts many perspectives ranging from anthropology to neurobiology to philosophy. The various research paradigms produce not only different theories and models of different aspects of design, but also different types of theories and models. For a quarter of a century, our research laboratory has explored design from the perspectives of artificial intelligence, cognitive science, and human-centered computing. Design research in these paradigms produces information-processing theories and computational models of aspects of design, as well as computer programs that implement and test the theories and models. These products in turn often form the basis for the development of interactive technologies for supporting aspects of design practice as well as pedagogical techniques for teaching elements of design theory and methods.

We have three main goals in this chapter. First, we want to briefly describe the perspectives of knowledge-based artificial intelligence, computational cognitive science, and human-centered computing, and in particular, the types of theories, models, programs, and tools they produce. Second, we want to illustrate some of the methods and artifacts of our research through a case study of problem–solution coevolution in biologically inspired design. Starting with the extant Structure-Behavior-Function model for expressing knowledge of technological and biological systems, we develop a knowledge model of design problems called SR.BID that is grounded in empirical data about biologically inspired design practice. Third, we want to present the SR.BID model that captures problem descriptions as well as problem–solution relationships in biologically inspired design. SR.BID forms the basis for ongoing development of new interactive tools for supporting biologically inspired design practice as well as new pedagogical techniques for learning about problem formulation.

A. K. Goel (✉) · M. E. Helms
Design and Intelligence Laboratory, School of Interactive Computing,
Georgia Institute of Technology, Atlanta, GA 30308, USA

A. Chakrabarti and L. T. M. Blessing (eds.), *An Anthology of Theories and Models of Design*, DOI: 10.1007/978-1-4471-6338-1_20,

20.2 Artificial Intelligence, Cognitive Science, and Human-Centered Computing

Artificial intelligence has several research paradigms. In this work, we are interested in the paradigm of knowledge-based artificial intelligence that has twin goals [22, 31]: to computationally understand human intelligence and to build intelligent systems with human-level intelligence. Theories and models in knowledge-based artificial intelligence typically use knowledge constructs to unify memory, reasoning, and learning processes, and thus address issues concerning the content, representation, organization, use, and acquisition of knowledge.

We are interested in computational cognitive science that seeks to computationally understand animal cognition [43]. A classical paradigm in computational cognitive science is human information processing that seeks to understand human behavior in terms of information processing in the human mind [37]. Another paradigm popular in modern cognitive science is situated cognition [6, 10] that seeks to understand human behavior in terms of interaction with the physical and social worlds.

Human-centered computing is an emerging interdiscipline within modern computing [29]. Human-centered computing takes human experience and its sociocultural context into consideration in the design of computational artifacts. In practical terms, human-centered computing is the next stage in the evolution of human–computer interaction as a discipline. As Fig. 20.1 shows, we are interested in human-centered computing at the intersection of artificial intelligence, cognitive science, and human–computer interaction. In particular, we are interested in research on artificial intelligence and cognitive science that produces interactive tool for supporting human designers in their work. Although not shown in Fig. 20.1, we are also interested in research on artificial intelligence, cognitive science, and human-centered computing that results in pedagogical techniques for teaching and learning design theory and methods.

20.3 Information-Processing Theories, Computational Models, Computer Programs, and Interactive Tools

We use the terms "theory" and "model" here in the sense of a scientific theory and a scientific model [11, 12, 32, 36]. A scientific theory is (i) based on testable hypotheses and makes falsifiable predictions, (ii) internally consistent and compatible with extant theories, (iii) supported by evidence, and (iv) modifiable as new evidence is collected. An important cognitive feature of a scientific theory is that it suggests a process or method for building, evaluating, revising, and accepting (or abandoning) a theory.

As indicated above, we are interested in information-processing theories of design. As an example, for a quarter of century the design research community has

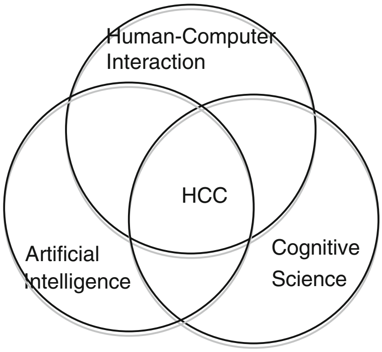

Fig. 20.1 Human-centered computing (HCC) at the intersection of artificial intelligence, cognitive science, and human–computer interaction

been developing Information-processing theories of analogical design (e.g., [17, 21, 35, 49]). In our own earlier work on analogical design, we have developed normative artificial intelligence theories, techniques, and tools for analogical design ranging from case-based design [18, 19] to cross-domain analogies [3, 20]. These theories are based on testable hypotheses about case-based design and cross-domain analogies in design, respectively, and some of their predictions have been evaluated through computational and experimentation.

A scientific model is an interpretation of a target system, process, or phenomenon that proposes or elaborates on the processes and mechanisms that underlie it. Like scientific theories, scientific models too have important cognitive features. First, models are abstractions of reality. They productively constrain reasoning by simplifying complex problems and thus suggest a course of analysis. Second, models are cognitive tools for generating explanations. They serve as tools both for specifying and organizing the current understanding of a system and for using that understanding for explanation and communication.

We are interested in two closely related kinds of models in design. First, we are interested in knowledge models. A knowledge model in design provides an ontology (i.e., a vocabulary) for representing the knowledge and a structure for organizing the knowledge in a design domain. For example, in our work on case-based design, we developed the Structure-Behavior-Function (SBF) knowledge model of the working of technological systems. The SBF knowledge model provides an ontology for expressing the knowledge of the system and a schema for organizing the knowledge [23, 24, 38]. The SBF model enables retrieval, adaptation, evaluation, and storage of design cases in addressing new design problems [19]. Similarly, in our work on cross-domain analogies in design, we developed Behavior-Function (BF) abstractions of SBF models that provide a vocabulary for representing teleological design patterns. The BF design patterns enable cross-domain analogies in designing new technological systems [3, 20].

Second, we are interested in computational models of design. While an information-processing theory of design is based on testable hypotheses and makes

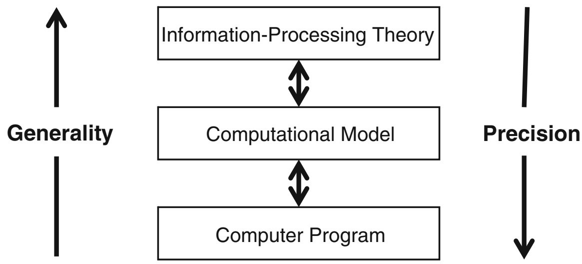

Fig. 20.2 Relationship among information-processing theories, computational models, and computer programs of design

falsifiable predictions, a computational model of design provides architectures, algorithms, and knowledge models for the theory. As Fig. 20.2 shows, computational models are more detailed and precise than information-processing theories. Thus, our computational model of cross-domain analogies in design [3, 20] provides an architecture that integrates memory, reasoning, and learning processes, SBF knowledge models of technological systems and BF knowledge models of design patterns, as well as algorithms for accessing, using, learning, and storing the design patterns.

The artificial intelligence paradigm also develops computer programs. A computer program is an experiment that implements the computational model and evaluates the information-processing theory. A computer program adds enough detail and precision to the computational model to be executable on a computer, as shown in Fig. 20.2. Thus, the Kritik [18, 19] and the Ideal [3, 20] computer systems implement our computational models and evaluated the information-processing theories of case-based design and cross-domain analogies, respectively.

The paradigm of human-centered computing also develops interactive technologies for supporting design practice. Indeed, interactive technologies have revolutionized design practice over the last generation, and insofar as we can see into the future, this trend likely will continue.

20.4 Problem–Solution Coevolution in Biologically Inspired Design: An Illustrative Case Study of Knowledge Modeling

The perspectives of artificial intelligence, cognitive science, and human-centered computing on design are mutually compatible. Thus, a design researcher can move from one paradigm to another depending on the research goal and the design context.

Further, knowledge models are common to all three paradigms. However, our discussion of knowledge models so far has been quite general and abstract. We now illustrate knowledge modeling through a case study of problem–solution coevolution in biologically inspired design (also known as biomimicry, biomimetics, and bioinspiration) [1, 2, 46, 48]. Over the last decade or so, the design research community has been studying biologically inspired design from the perspectives of artificial intelligence, cognitive science, and human-centered computing (e.g., [7, 41]). Our own interest in biologically inspired design spawned in part because it entails cross-domain analogies from biological systems to technological systems and thus provides an arena for further exploration of analogical design.

However, our work on biologically inspired design differs from our earlier work on analogical design in three fundamental ways. First, unlike the earlier normative artificial intelligence theories and models, our new work develops cognitive, descriptive theories, and models of analogical design (e.g., [28, 44]). Second, our work now has the additional goal of using our theories and models to develop interactive technologies (e.g., [25]; http://dilab.cc.gatech.edu/dane/) and pedagogical techniques for aspects of design. Third, our empirical studies have found that biologically inspired design entails not only cross-domain analogies but also problem–solution coevolution [26, 27]. Problem–solution coevolution is a well-known characteristic of creative design [14, 15, 33], but, insofar as we know, biologically inspired design has not been previously studied as entailing problem–solution coevolution. In traditional problem solving, the problem remains fixed even as solutions to the problem are generated. In problem–solution coevolution, the problem evolves as solutions are generated, with the current problem formulation influencing solution generation, and the current candidate solutions influencing problem formulation. Perhaps more interestingly, we found that biological analogies not only help generate solutions to a design problem, but also support inception and evolution of design problems [26, 27].

As much as the scope, focus, and methodology of our work have evolved over the years, our emphasis on grounding design processes in design knowledge has remained constant. The question then becomes what is a good knowledge model that can capture problem–solution coevolution in biologically inspired design? As one might expect, different researchers in biologically inspired design have developed different knowledge models, depending on the goal, scope, focus, and methodology of their work. Thus, Biomimicry 3.8 Institute has developed an ontology of functions of biological systems that purports to support its design model for generating design solutions [4]. Vincent and his colleagues have developed an ontology of biological systems that promises to support a TRIZ-like model of biologically inspired design [45]. Stone, McAdams and their colleagues have proposed the use of the extant function-flow ontology of Functional Basis for the task of concept generation in biologically inspired design [34]. Chakrabarti and his colleagues have developed a detailed SAPPhIRE knowledge model to support biologically inspired design [40]. All these knowledge models are normative, even if some of them are based on notions of best practices in biologically inspired design. Perhaps more importantly, all these models focus on design ideation in conceptual design (and thus do not address problem–solution coevolution).

In contrast, in this work we are interested in developing a knowledge model of design problems that can capture the process of problem–solution coevolution in biologically inspired design. We start with textual data from the practice of biologically inspired design in an educational setting and then derive the knowledge model of design problems called SR.BID. We validate the SR.BID model through comprehensive and repeatable categorization of unstructured textual data collected in the biologically inspired design practice.

20.5 Methodology and Data

Since 2006, we have observed ME/ISyE/MSE/PTFe/BIOL 4740, an interdisciplinary, project-based class taught yearly and jointly by biology and engineering faculty at Georgia Institute of Technology. In this course, mostly senior-level design students work in small interdisciplinary teams of 4–5 on open-ended design projects over the course of a semester. The extended, collaborative design projects typically involve identification of a design problem of interest to the team and conceptualization of a biologically inspired solution to the identified problem. Yen et al. [47] describe the course and the design projects in detail.

We use three data sets collected from observations of the design projects in the biologically inspired design class. The first set of data consisted of the project submissions of one design team in Fall 2008 that focused on capture of solar energy for use in homes. The project was selected as a typical example of biologically inspired design. The data consisted of four individual problem description assignments, a team mid-term presentation, and the team final presentation. We shall refer to this as the 2008 data set. The following is an excerpt from a problem description:

> I think this is a big gap between the static and fragile solar panels that we have so far engineered. So far, most solar panels are set up on a grid basis acting together especially when moving to the sun rather than as individual. Continuing off that tangent I think it would be interesting to have an individual solar panel that can stand alone and still function. The snail shell structure is stand alone and has the ability to passively dissipate heat by using the heat gradient so that it is cooler within the shell than the outside air and ground this would be helpful for allowing the interior of a structure with solar panels to remain cool.

The second set of data consisted of individual assignments given to students in Fall 2010, and collected in the third week of class. This assignment asked students to provide a short 1–2 page design problem description suitable for the biologically inspired design context. A total of 38 assignments were collected (one of which was eliminated as it belonged to a member of our research laboratory who was taking the class at the time). We shall refer to this as the Week 3 2010 data set.

The third set of data consisted of an individual assignment given to students in Fall 2010 and collected during the eighth week of class. This assignment consisted of problem descriptions between one quarter of a page and one full page in length. A total of 32 assignments were collected (the assignment from the member of our laboratory was again eliminated). We shall refer to this as the Week 8 2010 data set.

To analyze these data sets, we used a variation on the methodology of Grounded Theory [16, 42]. In the Grounded Theory methodology, a theory about any phenomenon is derived (solely) from data. In a recent variation, the theory is derived from data but the initial coding scheme is seeded with a predefined ontology [30]. As indicated above, we use the SBF knowledge model as a seed, and then derive the SR.BID model from the data about biologically inspired design.

20.5.1 Brief Review of the SBF Knowledge Model

SBF is a family of knowledge models that includes not only SBF models of biologically and technological systems, but also BF models of design patterns (as well as other models not described here) [19, 38]. Here, we briefly summarize the basic SBF model that consists of three nested high-level schemas, the *structure*, *behavior*, and *function* schemas [23]. The structure schema consists of a set of elements, which may be classified as *elements* such as *substances* or *components*, and *connections* among them. *Elements* may have associated *properties* and *values*, while *connections* express the relationship type (e.g., hinged) between *elements*.

The *behavior* schema consists of *states* and *transitions* between the *states*. *States* consist of a set of *elements*, and a set of *property—value* for the *element*. Each *transition* is annotated by *causal explanations* for the *transition.* Since one kind of *causal explanation* pertains to a *function* of a *component, behaviors* act as indices to *functions* of *components.*

The *function* schema consists of a *given* or *prerequisite state*, and one or more *makes* or *resultant states*. It also specifies one or more external *stimuli*. Also, it specifies the *behavior* that accomplishes the *function.* Thus, *functions* act as indices to *behaviors. Functions* can be of several types including *accomplishment, maintenance, prevention,* and *negation.*

20.5.2 Construction of the SR.BID Knowledge Model

We started with a single coder to map the problem description text data in the 2008 data set to concepts in the SBF knowledge model. During initial coding, our goal was to align the SBF ontology with the data and add new conceptual categories as they emerged from the data.

Figure 20.3 shows SR.BID's high-level ontology that emerged from our analysis. The ontology consists of six main concepts: *function, performance criteria, solution, deficiencies/benefits, constraints/specification,* and *operating environment*. *Solution*

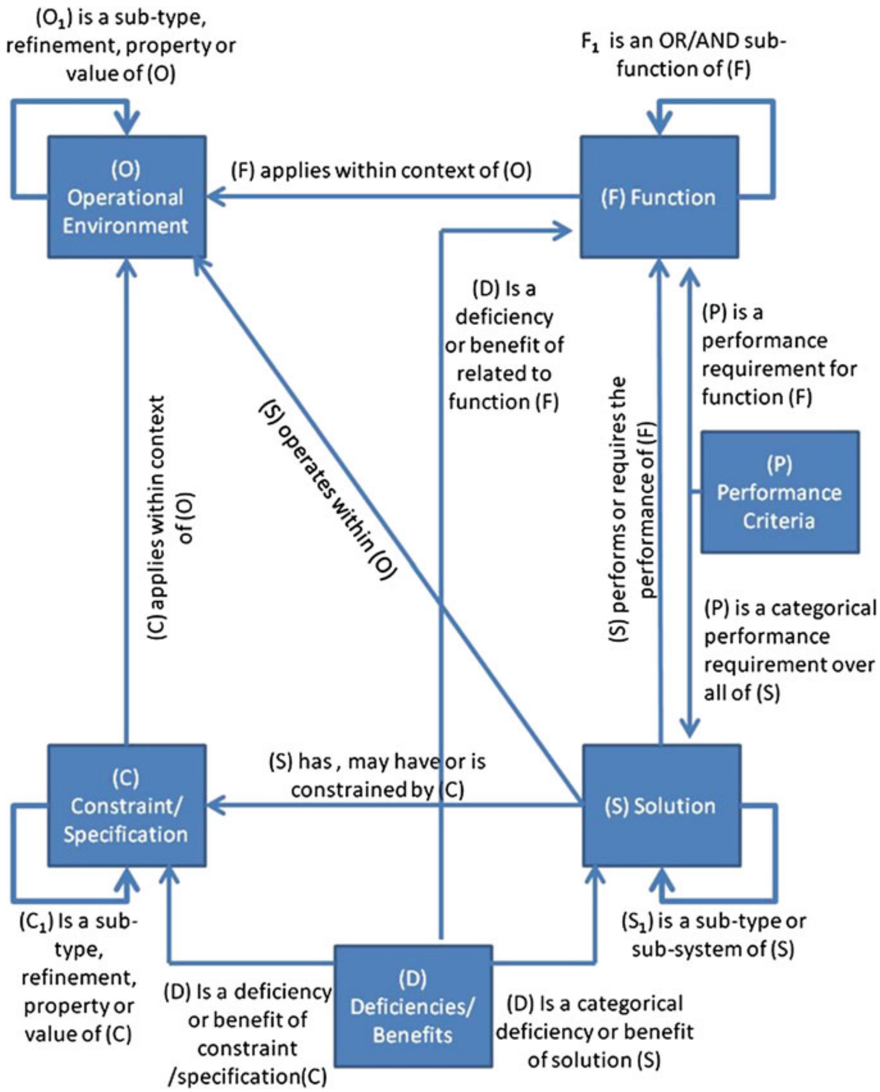

Fig. 20.3 The problem schema in SR.BID including both the main concepts and relationships

here refers to existing systems for achieving the given *function*, and *deficiencies/ benefits* pertain to negative/positive assessments of the *solution. Performance criteria* act as qualifiers on the Function (e.g., dissipate heat *passively*), and *constraints/ specification* describe constraints on the *solution* (e.g., cost).

20.5.3 Refinement of the SR.BID Model

Following the construction of the initial SR.BID model, we used two coders to refine and validate the model using the Week 3 2010 data set, which consisted of 37 design problem statements between one and two pages in length. The first coder was an author on this chapter (Helms) and was well versed with the coding process. The second coder was a third year undergraduate biology student new to the field of biologically inspired design, and without prior background knowledge in design or cognition, SBF, or SR.BID. We allocated half of the data (17 problem statements, selected at random) to training and refinement and used the remaining to draw samples for testing and validation.

This phase led to the identification of relationships among the six concepts in SR.BID's problem model, as shown in Fig. 20.3. This phase also led to identification of additional subcategories of the six categories in the model. Appendix 1 (Detailed Description of the SR.BID Knowledge Model), provides a complete listing and description of each category and subcategory. Note that as required of a knowledge model, the SR.BID model of design problems provides both an ontology for representing knowledge of design problems and a schema for organizing the knowledge, which allows capture of descriptions of specific problems such as the one on page 420.

After two passes on refinement and training, a random sample of five was pulled from the remaining problems to be used for validation. Each coder independently coded each test sample. We found the Cohen's Kappa measure of inter-coder reliability that adjusts for chance agreement to be 0.778. (Generally Cohen's Kappa values close to and above 0.8 are deemed acceptable.) After initial comparison, the two coders entered a negotiation phase, in which they attempted to resolve coding discrepancies. As expected, post-negotiation agreement levels were at significantly higher Cohen's Kappa values: 0.962 of concepts and 0.976 for relationships.

20.5.4 SR.BID Validation

To further test the conceptual soundness and potential usefulness of SR.BID, we applied it to the 2010 Week 8 data set, consisting of 31 brief problem statements. In this test we used a conservative dual-coding strategy over the entire data set. During dual-coding, each of the two coders is present during the session, and while one coder takes the lead, the second coder may question coding decisions leading to discussion and negotiation until a code is agreed upon. This ensures reliability much closer to the post-negotiated numbers shown in the previous test, with the additional cost of requiring two coders to code all documents. Tests of intra-coder reliability, conducted on the recoding of five problem statements selected at random 12 weeks after initial coding, demonstrate an agreement of 0.878 and 0.872 for coding concepts and relationships, respectively.

We found that the *function* concept is pervasive in most problem descriptions, occurring in 72.7 % of all conceptual relationships. The *solution* concept too is quite common, occurring in about half of the relationships. The *function-solution* relationship is the most common relationship, representing about one-fourth of all conceptual relationships in the observed sample. It is noteworthy that nearly 70 % of the *function-solution* relationships in our sample pertained to *existing* solutions rather than conjectured solutions. Understanding the role and influence of existing solutions such as biological analogs is of particular interest in biologically inspired design.

20.6 Discussion

In our perspective, knowledge models in design are intimately connected to information-processing theories and computational models of design tasks. As we study new design tasks, we develop new knowledge models appropriate to the task. Thus, as we studied memory, reasoning, and learning tasks in analogical design a generation ago, the SBF model logically evolved out of Chandrasekaran's Functional Representation [8, 9]: SBF representations supported the inferences required by the memory, reasoning, and learning tasks in analogical design. In a similar manner, as we study problem–solution coevolution in biologically inspired design, the SR.BID model of design problems is evolving out of the SBF model of the working of technological and biological systems.

The SR.BID model allows us to capture problem descriptions more deeply than the SBF model. In the basic SBF schema [23], a system's interaction with its external environment is captured in terms of system's *functions* and *external stimuli* from the environment to the system. Prabhakar and Goel [38] did describe the external and internal environments of a system but those ideas were not fully developed. SR.BID specifies *operational environment* explicitly. Similarly, *performance criteria* establish the metrics against which the *functions* of a design of a system may be evaluated. The frequency of occurrence of the *operating environment* and *performance criteria* concepts in our study seems to highlight their important role of problem formulation: they provide additional information needed to evaluate whether a *solution* satisfies the desired *function.* Dinar et al. [13] provide an alternative schema for representing problem descriptions.

As we noted above, the coded textual descriptions of biologically inspired design frequently refer to biological analogies and other existing solutions. This may have to do with the way in which design problem formulation occurs in biologically inspired design. Given a need, one method for problem formulation is to look to existing solutions that have been used to solve the need, or similar needs, in past. An existing solution provides a base case, a plan, or a pattern from which the designer might abstract key concepts, such as *functions*, which provide the points of traction necessary to begin formulating the design problem. This has deep implications for biologically inspired design because it shows that biological analogies may serve to help (re-)formulate problems as well as solve them.

20.7 Uses of SR.BID

Currently, we are using the SR.BID model in four ways. First, we are using it as a coding scheme to analyze additional data on problem–solution coevolution in biologically inspired design. In particular we are studying the influence of biological analogies on problem formulations and reformulations over time.

Second, we are using SR.BID as part of a pedagogical technique to help students in formulating design problems in the Georgia Tech ME/ISyE/MSE/PTFe/BIOL 4740 course on biologically inspired design. In past, problem formulation has been an extremely difficult task for students in the class [47]. In our pedagogical technique, students define their problems in terms of "four boxes:" *operational environment, function*, *constraints/specifications*, and *performance criteria.*

Third, we are developing an interactive technology for aiding students in evaluating cross-domain analogies in design. Designers in general lack a tool for systematic evaluation for cross-domain analogies. Thus, evaluation of analogies often is *ad-hoc*, and suffers from confirmation bias effects. Our tool uses the same "four-box" method to evaluate analogies in biologically inspired design. Students compare their four-box problem description against a four-box representation constructed for their biological analog, and then use this to frame a discussion of how their analogy is similar and dissimilar.

Finally, where most search engines for biologically inspired design focus on indexing by functions, we are using SR.BID to structure a knowledge base of design problems and biological systems to help facilitate search across the breadth of concepts found in the problem schema shown in Fig. 20.3.

20.8 Conclusions

Methodologies for research in design are receiving much needed attention (e.g., [5]). Our methodology for design research constructs information-processing theories, computational models, and computer programs of design. It also produces knowledge models, interactive tools, and pedagogical techniques for design.

Current information-processing theories of analogical design, including biologically inspired design, typically focus on use of analogy for generation of design ideas, and concepts for a given design problem. However, in tracing collaborative, extended, open-ended episodes of biologically inspired design we found that biological analogies often help not only in generating design ideas for a given formulation of the design problem, but also in (re-)formulating the problem itself. In fact, problem reformulation appears to have been the primary role of some biological analogies since the biological systems were not part of either the preliminary or final design solutions.

Evaluating our information-processing theory of biologically inspired design requires the construction of a computational model that specifies the architecture, algorithms, and knowledge model for problem–solution coevolution, where the knowledge model specifies the ontology and the schema for representing and organizing knowledge of design problems. In this chapter, we focused on the knowledge model. In particular, we used the SBF schema for representing knowledge of biological and technological systems as a seed for developing the SR.BID schema for representing problem descriptions in biologically inspired design. The conceptualization of the SR.BID problem schema was data driven, and grounded in the verbal descriptions designers provided for their designing. As measured by standard tests of coder reliability and coverage, the SR.BID constructs seem to provide comprehensive and reliable encoding of the verbal descriptions of interdisciplinary design teams engaged in biologically inspired design.

The SR.BID problem schema allows us to capture the problem descriptions design teams construct in collaborative, extended, open-ended biologically inspired design; it also enables us to capture the relationships between the problem and the solutions, as well as systematically trace the influence of the problem on the solution and vice versa in problem–solution coevolution in biologically inspired design. The SR.BID problem schema forms the basis of both pedagogical techniques for teaching about problem formulation and interactive tools for assessing cross-domain analogies for addressing a given design problem.

Acknowledgments We are grateful to the instructors and students of the ME/ISyE/MSE/PTFe/BIOL 4740 classes from 2006 through 2012, especially Professor Jeannette Yen, the main instructor and coordinator of the class. We are grateful also for the contributions of Marika Shahid, Swaroop Vattam, and Bryan Wiltgen. We thank the US National Science Foundation for its support of this research through an NSF CreativeIT Grant (0855916, Computational Tools for Enhancing Creativity in Biologically Inspired Engineering Design), and an NSF TUE Grant (1022778, Biologically !nspired Design: A Novel Interdisciplinary Biology-Engineering Curriculum.

Appendix 1: Detailed Description of the SR.BID Knowledge Model

The following tables describe the ontology of the SR.BID knowledge model of design problems that emerged from analyzing problem statements in the Week 3 2010 and Week 8 2010 data sets. These tables refine the high-level ontology of concepts and relationships of Fig. 20.3.

Solution	Description
Primary type	
Biological	The solution is a naturally occurring biological component, organism, or system
Man-made	The designers refer to a system which someone already built or created, or for which they generated prototypes or specifications
New design solution	The designers who are working on the problem are conjecturing a new design (or a design they think is new) to solve the problem
Secondary type	
Sub-solution	A sub-solution consists of many parts that together perform a specific function within the context of a larger solution
Subtype	A subtype solution expresses a “kind-of” relationship with another solution

Function	Description
Primary type	
Accomplishment	The default function type, accomplishment functions change the state of the world in an intended way
Preventative	Preventative functions keep a state OR another function from occurring
Maintenance	Functions that maintain a state are considered maintenance for example “the thermostat regulates temperature” is a maintenance function
Allow	Allow functions enable a state OR another function to occur
Negation	Negative functions are stated as NOT performing another function, for instance this application does not produce light
Secondary type	
Sub-function, AND	When there are multiple sub-function relationships for a given function, AND-type relationships that specify that the related sub-functions must all be accomplished in order to achieve the parent function
Sub-function, OR	When there are multiple sub-function relationships for a given function, the OR-type relationship specifies that one of the functions must be accomplished to achieve the parent function

Operating environment	Description
Primary type	
Location	The places in which the system is intended to operate
Condition-qualitative	Qualitative conditions under which the system is intended to operate
Condition-quantitative	High/low-end values, expected values, or ranges
Time	The time during which the system must operate for example, “at night.” Words like “when,” “after,” “while,” “as,” and “during” are often used to express a temporal environment
User	The phrase describes an intended user or class of users for the system
Entity	The phrase describes an entity, often biological but sometimes technological, that interacts with the system
System	The phrase describes another system within which the system is intended to work or connect

Constraints and specifications	Description
Primary type	
Material	The material of which one or more components of the design will be composed
Information	Information can be in the form of energetic signals, bits and bytes, or may be encoded in the physical structure of a thing
Energy	Energy can be found throughout a system in many forms; the energy subtype is used when a specified form of energy is discussed within the confines of the system
Time	Includes timeframes not related to the operation of the design
Component	Includes descriptions of specific parts of a solution or design, or groups of parts
Property/value	Concerns the properties of the system as a whole or their values
Shape	Includes the shape of the components or of the design
Spatial orientation	These specify the spatial relationship or orientation between or among one or many components, systems, or subsystems
Structural relationship	Any phrase specifying which components are related by means of connecting joints and contacts points
Cost	Usually in monetary terms, but this could also be in terms of any resource of concern; absolute; or relative
Secondary type	
Limiting	Limiting specifications/constraints are those which require a designer to use a smaller subset of design elements
Enabling	Enabling specifications/constraints offer new possibilities for design elements without enforcing their use
Existing	Existing specifications/constraints discuss the specific properties of an existing design

Performance criteria	Description
Primary type	
Specific	States the specific value or range of the performance criteria
Relative	Uses comparative terms such are “quieter than solution X,” without explicitly stating the performance of the compared to solution
Actual	States the performance of an existing solution

Deficiency/ Benefit	Description
Primary type	
Deficiency	Deficiencies can relate to any element of an existing solution or proposed design, highlighting an unfavorable aspect of that element
Benefit	Benefits can relate to any element of an existing solution or proposed design, highlighting a favorable aspect of that element

References

1. Bar-Cohen Y (ed) (2011) Biomimetics: nature-based innovation. CRC Press, Boca Raton
2. Benyus J (1997) Biomimicry: innovation inspired by nature. William Morrow, New York
3. Bhatta S, Goel A (1997) Learning generic mechanisms for innovative strategies in adaptive design. J Learn Sci 6(4):367–396
4. Biomimicry 3.8 Institute (2008) AskNature. http://www.asknature.org/. Accessed on 25 Apr 2011
5. Blessing L, Chakrabarti A (2009) DRM: A design research methodology. Springer, London
6. Brown J, Collins A, Duguid P (1989) Situated cognition and the culture of learning. Educational Researcher 18(1):32–42
7. Chakrabarti A, Shu L (2010) Biologically inspired design. AIEDAM 24:453–454
8. Chandrasekaran B (1994) Functional representation: a brief historical perspective. Appl Artif Intell 8(2):173–197
9. Chandrasekaran B, Goel A, Iwasaki Y (1993) Functional representation as design rationale. IEEE Comput 26:48–56
10. Clancey W (1997) Situated cognition: on human knowledge and computer representations. Cambridge University Press, Cambridge
11. Clement J (2008) Creative model construction in scientists and students: the role of imagery, analogy, and mental simulation. Springer, Dordrecht
12. Darden L (1998) Anomaly-driven theory redesign: computational philosophy of science experiments. In: Bynum T, Moor J (eds) The digital phoenix: how computers are changing philosophy. Blackwell Publishers, New York, pp 62–78
13. Dinar M, Shaj J, Hunt G, Campana E, Langley P (2011) Towards a formal representational model of problem formulation in design. In: Proceedings of ASME 2011 IDETC/CIE Conference, Washington DC
14. Dorst K (2003) The problem of design problems. In: Cross N, Edmonds E (eds) Expertise in design. Creativity and Cognition Studio Press, Sydney
15. Dorst K, Cross N (2001) Creativity in the design process: co-evolution of problem-solution. Des Stud 22(5):425–437
16. Glaser B, Strauss A (1967) The discovery of grounded theory: strategies for qualitative research. Aldine, Chicago
17. Gebhardt F, Voß A, Gräther W, Schmidt-Belz B (1997) Reasoning with complex cases. Kluwer, Norwell, MA
18. Goel A (1992) Representation of design functions in experience-based design. In: Brown D, Waldron M, Yoshikawa H (eds) Intelligent computer aided design. North-Holland, Amsterdam, pp 283–308
19. Goel A, Bhatta S, Stroulia E (1997) Kritik: an early case-based design system. In: Maher M, Pu P (eds) Issues and applications of case-based reasoning in design. Mahwah, Erlbaum, pp 87–132
20. Goel A, Bhatta S (2004) Design patterns: a unit of analogical transfer in creative design. Adv Eng Inform 18(2):85–94
21. Goel A, Chandrasekaran B (1988). Integrating model-based reasoning with case based reasoning for design problem solving. In: Proceedings of AAAI-88 workshop on AI in design, Minneapolis
22. Goel A, Davies J (2011) Artificial Intelligence. In: Sternberg R, Kaufman S (eds) Cambridge handbook of intelligence, 3rd edn. Cambridge University Press, Cambridge
23. Goel A, Rugaber S, Vattam S (2009) Structure, behavior and function of complex systems: the SBF modeling language. Artif Intell Eng Des Anal Manuf 23:23–35
24. Goel A, Stroulia E (1996) Functional Device Models and Model-Based Diagnosis in Adaptive Design. Artif Intell Eng Des Anal Manuf 10:355–370

25. Goel A, Vattam S, Wiltgen B, Helms M (2012) Cognitive, collaborative, conceptual and creative—four characteristics of the next generation of knowledge-based CAD systems: a study in biologically inspired design. Comput Aided Des 44(10):879–900
26. Helms M (2011) Solution based problem evolution and problem inception in biologically inspired design. Technical report, GVU, Georgia Institute of Technology. GIT-GVU-11-10
27. Helms M, Goel A (2012) Analogical problem evolution in biologically inspired design. In: Proceedings fifth international conference on design computing and cognition, College Station, Texas, July 2012. Springer
28. Helms M, Vattam S, Goel A (2009) Biologically inspired design: product and processes. Des Stud 30(5):606–622
29. Kling R, Star SL (1998) Human centered systems in the perspective of organizational and social informatics. Comput Soc 28(1):22–29
30. Lamp J, Milton S (2007) Grounded theory as foundations for methods. In: Applied ontology, Proceedings of the 4th international conference on qualitative research in IT and IT in qualitative research (QualIT), Victoria University of Wellington
31. Langley P (2012) The cognitive systems paradigm. Adv Cognitive Sys 1:3–13
32. Machamer P, Darden L, Craver C (2000) Thinking about mechanisms. Philos Sci 67:1–25
33. Maher ML, Tang H (2003) Co-evolution as a computational and cognitive model of design. Res Eng Design 14(1):47–64
34. Nagel J, Nagel R, Stone R, McAdams D (2010) Function-based, biologically inspired concept generation. Artif Intell Eng Des Anal Manuf 24:521–535
35. Navinchandra D (1991) Exploration and innovation in design. Springer, New York
36. Nersessian N (2008) Creating scientific concepts. MIT Press, Cambridge, MA
37. Newell A, Simon H (1972) Human problem solving. Prentice-Hall, Englewood Cliffs
38. Prabhakar S, Goel A (1998) Functional modeling for enabling adaptive design of devices for new environments. Artif Intell Eng 12:417–444
39. Russell S, Norvig P (2010) Artificial intelligence: a modern approach, 3rd edn. PrenticeHall, Upper Saddle River
40. Sartori J, Pal U, Chakrabarti A (2010) A Methodology for supporting “transfer” in biomimetic design. AIEDAM 24:483–506
41. Shu L, Ueda K, Chiu I, Cheong H (2011) Biologically inspired design. CIRP Annals Manuf Technol 60:673–693
42. Strauss A, Corbin J (1990) Basics of qualitative research: grounded theory procedures and techniques. Sage, Thousand Oaks
43. Thagard P (2005) Mind: introduction to cognitive science, 2nd edn. MIT Press, Cambridge
44. Vattam S, Helms M, Goel A (2010) A content account of creative analogies in biologically inspired design. Artif Intell Eng Des Anal Manuf 24:467–481
45. Vincent J, Bogatyreva O, Bogatyrev N, Bowyer A, Pahl A (2006) Biomimetics: its practice and theory. J R Soc Interface 3:471–482
46. Vincent J, Mann D (2002) Systematic technology transfer from biology to engineering. Phil Trans R Soc Lond 360:156–173
47. Yen J, Weissburg M, Helms M, Goel A (2011) Biologically inspired design: a tool for interdisciplinary education. In: Bar-Cohen Y (ed) Biomimetics: nature-based innovation. Taylor & Francis, Boca Raton
48. Yen J, Weissburg M (2007) Perspectives on biologically inspired design: introduction to the collected contributions. J Bioinspir Biomim 2
49. Zhao F, Maher M (1988) Using analogical reasoning to design buildings. Engineering with Computers 4:107–122